

REMARKS

The Office Action rejected Claims 69 – 71 under 35 U.S.C. § 101 as being directed to non-statutory matter. Claim 69 was cancelled and Claims 70 and 71 were amended to overcome the rejection. Applicant respectfully requests that the rejection be withdrawn.

Support for amendments to Claims 48, 59, and 71 and the additions of new Claims 72 – 75 can be found, for example, on

- (a) Page 40, line 21 to Page 52, line 7,
- (b) Page 59, line 14 to Page 63, line 12,
- (c) Page 115, lines 19 – 25, and
- (d) FIGS. 16, 23A, 23B, 23C, and 56.

For example, the claim language exemplified in Claim 48 “one of the plurality of programs includes a command for branching which instructs a playback device to perform branching based on a title number” refers to the command Jump Title#m which indicates that the playback device should perform branching based on the title number m, such as “000.MOVIE” as shown in FIG. 23B. (Pg. 41, lns. 8 – 23, Pg. 63, lns. 1 - 12)

The present invention is directed towards a system of interactive mediums and devices which encourages programmers to produce interactive programs for inclusion in the interactive medium and distributors to distribute the interactive medium. (Pg. 6, lns. 8 – 14, Pg. 13, lns. 5 – 20) The present invention accomplishes this by using, for example an object-oriented programming language such as JAVATM. The use of an object-oriented programming language such as JAVATM allows one program to be created which can be used on multiple platforms. (Page 43, lns. 12 to 25) Since the programmer only has to create one program, the programmer’s developmental costs are reduced. This can lead to more innovation by the programmer.

Furthermore, the present invention utilizes branch commands by referencing a table instead of directly referencing the enhanced mode. Advantageously, this allows a new table to be downloaded whenever alterations are needed to the programming order, such as when certain objectionable scenes should be removed or if the user does not have the appropriate software to view the enhanced material. Thus, instead of reordering the material on the recorded medium or performing a recall of the recorded medium, the new table can be downloaded and referenced by the playback device. Increased operational assurances will provide creative assurances to the distributors as they will have greater assurances that the movies they create will be viewed in the proper environment. (Pg. 9, ln. 9 – Pg. 11, ln. 7, Pg. 13, lns. 5 – 20) The use of the table is also advantageous for security and copyright reasons as listed on Page 11, line 16 to Page 13, line 4. Increased security and copyright assurances will also provide monetary incentives for distributors to create movies as they will have greater assurances that they will be able to recoup their costs or prevent their movies. (Pg.11, ln. 8 – Pg. 12, ln. 4)

The Office Action rejected Claims 48 – 71 under 35 U.S.C. § 102(e) as being anticipated by *Cho et al.* (U.S. Patent Pub. No. 2002/0176693) in reference of *Saeki et al.* (U.S. 6,067,400).

It appears that the Office Action is using both the *Cho* and the *Saeki* reference in an obviousness-type rejection under 35 U.S.C. § 103. If, however, Applicant is wrong in its assumption and the Office Action is instead using *Cho* and the *Saeki* individually in an anticipatory manner under 35 U.S.C. § 102, Applicant requests the opportunity to address such a rejection.

Cho is directed towards a recording medium in which the moving picture data is recorded together with the script files which have additional information relating to the moving picture data. (¶ 0002)

Saeki is a patent previously granted to the Applicant. *Saeki* is directed towards a multimedia optical disc and reproduction apparatus which allows for highly-responsive, highly-interactive, and efficient reproduction of an application using a simplified menu operation by the user. (Col. 3, lns. 62 – 66) It accomplishes this by restricting storage of the menu image to the first sub-area which contains the moving picture data which is played at the same time as the menu image. Furthermore, the control information including button control data for responding to a user's operation is stored in a second sub-area adjacent to the first sub-area. This allows for quick access to the menu operation. (Col. 4, lns. 10 – 24) In addition, *Saeki* also teaches forcibly activating one of the menu buttons after a period of time to ensure that the movie is not interrupted while waiting for the user to select the menu button. (Col. 4, lns. 34 – 52)

Cho and *Saeki* do not teach or suggest “one of the plurality of programs includes a command for branching which instructs a playback device to perform branching based on a title number.” Cho utilizes a link data file table shown in FIG. 6 to match a program chain information (“PGCI”) to a link data file. (¶ 0039) Thus, a link data file may have a filename indicative of an associated PGCI to distinguish what link data file is associated with what PGCI. For example, the first link data file associated with PGCI “PGCI #1” is named “PGC_1.lk.” Likewise, the second link data file associated with PGCI “PGCI #2” is named “PGC_2.lk.” As can be seen, PGC_1 and PGC_2 are character names and not title numbers since the titles are not in a purely numerical format. (¶ 0039)

Saeki does not disclose the formatting of the branching commands, as shown in FIGS. 16 and 17. *Saeki* merely indicates what the branching commands are, but does not indicate their formatting.

In contrast, in the present invention, the branching command instructs the playback device to perform branching based on a numerical title as shown in FIG. 23C. In FIG. 23C, the naming convention for a branch destination is ZZZ.CLASS, where ZZZ is a numerical title. (Pg. 60, ln. 11 – Pg. 61, ln. 4) The ZZZ is a 3-digit identification number appended to the individual dynamic scenarios which uniquely identifies the dynamic scenarios. (Pg. 49, ln. 22 – Pg. 50, ln. 9) In FIG. 24A, ZZZ.CLASS can be, for example, 001.CLASS.

Cho and *Saeki* also fail to disclose

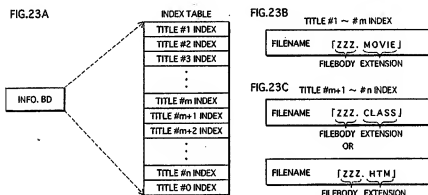
[T]he table comprises (a) combinations of program identification information and mode information and (b) title numbers corresponding to the combinations, the program identification information instructs the playback device on a program to be executed for dynamic control, when branching to a title of a corresponding title number, and the mode information shows whether the program to be executed for dynamic control is a program described in an object-oriented programming language.

Cho does not disclose the title information corresponding to the mode information. In *Cho*, the table shown in FIG. 6 merely corresponds program chain information to the link. The PGC_1 in PGC_1.lnk, however, is not the title of the program to be executed for dynamic control. PGC_1 is the program chain information. While the program chain information may contain a list of programs, the program chain information is not the title of the program. Likewise, the Video Title Set Video Object Unit Address Map (VTS_VOBU_ADMAP1) also does not show corresponding between the title of the program and the mode information as it is only the start address of a video object unit in a Video Title Set Object Unit. (¶ 0034, FIG. 3)

Saeki also does not remedy the deficiencies of *Cho*. *Saeki* discloses a title search pointer table which is an index of the PGCs stored in the PGC information table. The title search pointer table stores pointers for the storage position for the PGC which should be executed first for each

title, for example, a pointer which shows the storage position of the PGC information that shows the leading PGC for the interactive movie. (Col. 16, lns. 36 – 41). While the location of the first PGC to be executed may be stored, the title search pointer table does not indicate which mode the PGC should be executed in. Furthermore, any command for branching in *Saeki* is done through button commands set for every button in the button information. The button information is part of management pack incorporated in a video object equivalent to video data. Thus, *Saeki* does not instruct a playback device on a program to be executed for dynamic control, when branching to a title of a corresponding title number. (Col. 15, ln. 53 - Col. 16, ln. 18)

In contrast, in the present invention, INFO.BD-ROM includes an Index Table as shown in FIGS. 23A, FIG. 23B, and FIG. 23C reproduced below:



As seen in FIG 23B, the Title #1 to Title #m INDEX each has a filename which includes a filebody and an extension. (Pg. 60, lns. 1 – 4) The filename is indicated by a title number ZZZ which is a numerical number such as 001 and an extension such as MOVIE. (Pg. 60, lns. 1 – 2) The program identification is the title number ZZZ, which identifies the dynamic scenario to be executed. The extension is the mode information, which indicates the type of dynamic scenario

to be executed. In FIG. 23B, the extension is MOVIE which indicates that the dynamic scenario to be executed should be executed in a MOVIE mode. (Pg. 72, lns. 16 – 18) However, in FIG. 23C, the extension CLASS indicates that the dynamic scenario to be executed should be in an object-oriented programming language and should be executed, for example, in a JAVA mode. (Pg. 72, lns. 18 – 20)

Cho and *Saeki* also do not teach or suggest

[T]he mode information shows whether the program to be executed for dynamic control is a program described in an object-oriented programming language.

The Office Action on pages 3 and 4 cite to the script-flag in *Cho* as the mode information. The script flag in *Cho* indicates whether there is a stored script file linked with the selected program chain. (¶ 0058) However, the script file is written in the conventional Hyper-Text Markup Language (HTML), which the present application indicates is different from an object-oriented programming language. (¶ 0031) Thus, *Cho* does not teach that the script file to be executed for dynamic control is a program described in an object-oriented language. Furthermore, since the script file is not an object-oriented language, *Cho* does not teach the mode information either since the script-flag merely indicates the presence of the script file.

Saeki also does not remedy the deficiencies of *Cho*. In *Saeki*, the PGC information management table is cited on Page 3 of the Office Action as the program to be executed. The PGC information management table, however, only indicates the reproduction route of the VOBs. That is, the PGC information management table indicates the order in which the VOBs are produced. (Col. 16, lns. 42 – 50) However, there is no indication that the PGC information is in an object-oriented language. Since the PGC information is not an object-oriented language, *Saeki* also does not disclose the “mode information.”

In contrast, in the present invention, the mode information is shown in FIG. 23A, 23B, and 23C, the mode information is described in the extension. (Pg. 59, ln. 18 – Pg. 60, ln. 10) The mode information is MOVIE when the branch-destination dynamic scenario is the MOVIE mode. (Pg. 72, lns. 16 – 18) The mode information is CLASS when the branch-destination dynamic scenario is the Java™ mode, where Java™ is an object-oriented programming language. (Pg. 72, lns. 18 – 20, Pg. 43, lns. 12 - 14) The mode information is HTML, when the branch-destination dynamic scenario is the Browser mode, where HTML is different from an object-oriented programming language. (Pg. 72, lns. 20 – 22) Thus, the extension or mode information reveals whether mode switching is necessary. (Pg. 72, lns. 22 – 23)

Applicant submits that any combination of references that must be modified beyond their functions is suggestive of an unintended use of hindsight that may have been utilized to drive the present rejection. This is particularly true for an Examiner who is attempting to provide a diligent effort that only patentable subject matter occurs. The KSR Guidelines do not justify such an approach. There is still a requirement for the Examiner to step back from the zeal of the examination process and to appreciate that a Patent Examiner has to wear both hats of advocating a position relative to the prior art while at the same time objectively rendering in a judge-like manner a decision on the patentability of the present claims.

As set forth in MPEP 2142,

To reach a proper determination under 35 U.S.C. §103, the examiner must step backward in time and into the shoes worn by the hypothetical “person of ordinary skill in the art” when the invention was unknown and just before it was made. In view of all factual information, the examiner must then make a determination whether the claimed invention “as a whole” would have been obvious at that time to that person. Knowledge of applicant’s disclosure must be put aside in reaching this determination, yet kept in mind in order to determine the “differences,” conduct the

search and evaluate the “subject matter as a whole” of the invention. The tendency to resort to “hindsight” based upon applicant’s disclosure is often difficult to avoid due to the very nature of the examination process. However, impermissible hindsight must be avoided and the legal conclusion must be reached on the basis of the facts gleaned from the prior art.

All arguments for patentability with respect to Claim 48 are repeated and incorporated herein for Claims 59, 70, and 71.

With respect to Claim 72 through 75, neither *Cho* nor *Saeki* disclose the use of the Java™ language. As previously noted, *Cho* indicates that the script files are in HTML language. Likewise, there is no mention of the use of the JAVA™ language in *Saeki*.

In contrast, the present invention uses the JAVA™ language. As the specification notes on Page 43, lines 12 to 25:

As with C++, the Java language is an object-oriented programming language. The difference from C++ is that, while C++ is implemented in operating systems, the Java language defines Java Virtual Machines and is implemented in main operating systems, such as Windows and Linux. Therefore, the use of the Java language allows descriptions of processing procedures independent of operation systems. . . . [E]ven if the execution environments differ from maker to maker, the Java language is able to describe execution environment-independent processing procedures.

Thus, the use of the JAVA™ language can simplify development with operational assurances for the programmer. The programmer can now produce one program which can work on a variety of playback devices instead of having to produce multiple programs for each version of a playback device. With a lower cost of development, programmers will be encouraged to produce programs for the DVD.

Claims 72 – 75 depend from and further define independent Claims 48, 59, 70, and 71 and are thus allowable, too.

If there are any questions with regards to this matter, the undersigned attorney can be contacted at the listed phone number.

Very truly yours,

SNELL & WILMER L.L.P.

A handwritten signature in black ink, appearing to read 'J. Price', is written over a horizontal line.

Joseph W. Price
Registration No. 25,124
600 Anton Boulevard, Suite 1400
Costa Mesa, California 92626-7689
Telephone: (714) 427-7420
Facsimile: (714) 427-7799